

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 1 030 493 A1

(12) DEMANDE DE BREVET EUROPEEN

(43) Date de publication:
23.08.2000 Bulletin 2000/34

(51) Int Cl.7: H04L 29/06

(21) Numéro de dépôt: 00400366.1

(22) Date de dépôt: 09.02.2000

(84) Etats contractants désignés:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Etats d'extension désignés:
AL LT LV MK RO SI

- Lattmann, Joel
77420 Champs sur Marne (FR)
- Bonjour, Servane
92100 Boulogne Billancourt (FR)
- Guesdon, Hervé
92240 Malakoff (FR)

(30) Priorité: 12.02.1999 FR 9901714

(71) Demandeur: FRANCE TELECOM
75015 Paris (FR)

(74) Mandataire: Loisel, Bertrand
Cabinet Plasseraud,
84, rue d'Amsterdam
75440 Paris Cédex 09 (FR)

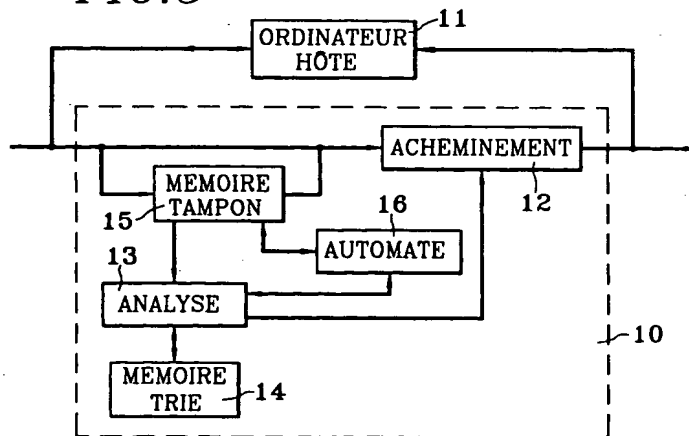
(72) Inventeurs:
• Duret, Christian
92320 Chatillon (FR)

(54) Procédé pour associer des références d'acheminement à des paquets de données au moyen d'une mémoire trié, et dispositif de traitement de paquets appliquant ce procédé

(57) On analyse successivement, à partir de registres portiers différents de la mémoire TRIE, différentes portions d'un en-tête de chaque paquet contenant des informations de protocole. A l'arrivée d'un paquet, on stocke son en-tête dans une mémoire tampon (15) et on analyse une première portion de l'en-tête stocké. Chaque analyse d'une portion de l'en-tête produit soit la référence d'acheminement associée au paquet, soit une référence intermédiaire comportant un premier code permettant de localiser, à un emplacement arbitraire

de la mémoire tampon, une portion suivante à analyser et un second code permettant de localiser, à un emplacement arbitraire de la mémoire TRIE un registre portier à partir duquel cette portion suivante est à analyser. Après avoir analysé la première portion d'un en-tête stocké, on en analyse des portions suivantes conformément aux premiers et seconds codes contenus dans les références intermédiaires successivement produites, jusqu'à ce que la référence d'acheminement soit produite.

FIG.3



EP 1 030 493 A1

BEST AVAILABLE COPY

Description

[0001] La présente invention concerne les mémoires associatives, et particulièrement les mémoires de type « TRIE » (du verbe anglais « reTRIEve »).

[0002] Le principe de la mémoire « TRIE » a été proposé par R. de la Briandais et E. Fredkin vers la fin des années 1950 (voir E. Fredkin et al.: « *Trie Memory* », Communications of the ACM, Vol.3, No.9, septembre 1960, pages 490-499). Il consiste à découper les chaînes de bits à reconnaître en tranches successives de longueur fixe (de K bits) et à les intégrer dans un tableau T à deux dimensions. Chaque ligne du tableau constitue un *registre* de 2^K cellules élémentaires. Un registre (R) est attribué à chaque tranche de la chaîne et on associe une cellule dans le registre à la valeur (V), comprise entre 0 et 2^K-1 de cette tranche. Le contenu ($C = T[R,V]$) de la cellule ainsi déterminée représente soit le registre attribué à la tranche suivante (ou *pointeur*), soit une référence de fin d'analyse (ou « *status* ») si l'analyse de la chaîne doit se terminer sur cette tranche.

[0003] Le registre attribué à la première tranche de la chaîne, qui est aussi le point d'entrée de la table, est appelé *portier*. Les données à analyser sous forme de chaînes de bits, c'est-à-dire à comparer avec le contenu de la mémoire TRIE, seront également nommées *routes* ci-après. On appellera *chemin* dans la table la succession des cellules chaînées associées à une route. Chaque registre de la table sera dit d'ordre ≥ 0 s'il est attribué à la (i+1)-ième tranche d'une ou plusieurs routes mémorisées. Le registre portier est donc d'ordre 0. La mémoire TRIE associe à chacun de ses registres d'ordre ≥ 0 une unique séquence de iK bits correspondant aux iK premiers bits de chaque route dont le chemin dans la table passe par une cellule du registre en question.

[0004] L'exemple suivant montre une représentation du stockage des données dans une mémoire TRIE dans le cas particulier où K=4. La valeur de chaque tranche est représentée par un digit en numérotation hexadécimale (0,1,...,E, F), et les registres contiennent chacun $2^4=16$ cellules.

[0005] Soit à reconnaître les routes qui débutent par les motifs 45A4, 45AB, 67AB, 788A et 788BD, auxquels on attribue respectivement les status S0, S1, S2, S3 et S0 (un même status peut être partagé par plusieurs routes). En portant le registre R en indice de ligne, la valeur V des tranches en indice de colonne, et en prenant comme portier le registre $R_0=0$, la table de la mémoire TRIE peut se présenter comme représenté sur la figure 1, où les données soulignées sont des status. Les motifs 45A4, 45AB, 67AB, 788A et 788BD sont respectivement représentés dans la table de la figure 1 par les chemins :

$T[0,4] \rightarrow T[1,5] \rightarrow T[2,A] \rightarrow T[3,4];$

$T[0,4] \rightarrow T[1,5] \rightarrow T[2,A] \rightarrow T[3,B];$

$T[0,6] \rightarrow T[4,7] \rightarrow T[5,A] \rightarrow T[6,B];$

$T[0,7] \rightarrow T[7,8] \rightarrow T[8,8] \rightarrow T[9,A];$

$T[0,7] \rightarrow T[7,8] \rightarrow T[8,8] \rightarrow T[9,B] \rightarrow T[10,D].$

[0006] On voit sur cet exemple que tous les motifs commençant par une partie commune de iK bits sont représentés par un début de chemin commun dans la mémoire, menant au registre d'ordre i auquel est associée la séquence formée par ces iK bits.

[0007] Si l'on considère une route à analyser, découpée en une suite de tranches binaires de valeurs V_i avec $0 \leq i \leq N$ et $\{R_i\}$ la suite des registres associés aux valeurs V_i , R_0 désignant encore le registre portier, l'algorithme d'analyse mis en oeuvre peut être celui représenté sur la figure 2.

[0008] A l'initialisation 1 de cet algorithme, le rang d'analyse i est mis à 0 et le registre portier R_0 est sélectionné comme registre R. Dans chaque itération de rang i, le contenu C de la cellule $T[R,V_i]$ désignée par la (i+1)-ième tranche V_i de la route dans le registre d'ordre i sélectionné est lu à l'étape 2. Si cette cellule contient un pointeur de poursuite d'analyse, ce qu'indique au test 3 la valeur 1 d'un bit $FP(C)$ mémorisé dans la cellule, le registre d'ordre i+1 désigné par ce pointeur $Ptr(C)$ est sélectionné comme registre R pour l'itération suivante à l'étape 4, et le rang i est incrémenté. Quand le test 3 révèle une cellule qui ne contient pas un pointeur ($FP(C)=0$), le status $Ref(C)$ lu dans la cellule concernée est retourné à l'étape 5 comme résultat de la consultation de la table.

[0009] Cet algorithme permet l'analyse de routes comportant un nombre quelconque de tranches. Une même table

peut être utilisée pour plusieurs types d'analyses en gérant les données à partir de portiers différents. De plus, il permet de maîtriser le temps d'analyse des données : l'analyse d'un nombre N de tranches de K bits durera au plus N fois la durée d'une itération.

[0010] L'algorithme de la figure 2 peut être mis en oeuvre de façon très rapide par un composant matériel gérant les accès au tableau de mémoire. Il permet notamment la réalisation de routeurs de haute performance pour des réseaux de télécommunications à commutation de paquets. L'en-tête des paquets est analysé au vol par le composant, et le statut associé à une route désigne par exemple un port de sortie du routeur vers lequel doivent être acheminés les paquets portant une adresse de destination conforme à cette route.

[0011] Un tel routeur peut être multi-protocoles. Pour cela, on analyse des portions différentes de l'en-tête à partir de portiers différents. Par exemple, une première analyse d'un (ou plusieurs) champ de l'en-tête désignant le protocole employé et/ou la version de ce protocole peut être analysée à partir d'un premier portier. Cette première analyse fournit une référence qui, bien que correspondant à une fin logique d'analyse, peut être matérialisée dans la mémoire TRIE par un pointeur de poursuite d'analyse désignant un autre registre portier à utiliser pour analyser la suite de l'en-tête. La référence en question peut également déclencher des temporisations ou des sauts d'un nombre déterminé de bits dans l'en-tête analysé afin de pouvoir choisir quelle portion de l'en-tête doit être ensuite analysée. Dans la pratique, un certain nombre d'analyses sont généralement exécutées successivement, pour déclencher les opérations requises par les protocoles supportés en fonction du contenu des en-têtes. L'une de ces analyses portera sur l'adresse de destination pour accomplir la fonction de routage proprement dite.

[0012] Un routeur du type ci-dessus est décrit dans le brevet français 2 707 775. Au sujet de l'utilisation de mémoires TRIE dans des routeurs, on pourra se reporter à l'article « *Putting Routing Tables in Silicon* », de T.B. Pei et al., IEEE Network Magazine, janvier 1992, pages 42-50.

[0013] Le fait de pouvoir enchaîner plusieurs analyses élémentaires avec entre elles des sauts programmables procure une grande souplesse au procédé, particulièrement pour le traitement de protocoles encapsulés suivant plusieurs couches du modèle ISO. L'analyse au vol des tranches de l'en-tête au fur et à mesure de leur arrivée procure par ailleurs une grande rapidité.

[0014] Néanmoins, dans un certain nombre de cas de figures, il est utile de revenir en arrière dans l'en-tête analysé pour examiner certains champs dans un ordre différent de l'ordre d'arrivée. Ceci permet fréquemment une meilleure optimisation de la taille mémoire requise. C'est par ailleurs rendu nécessaire par certains protocoles, tels que le protocole de réservation RSVP ou des protocoles de multidiffusion (multicast).

[0015] Un but de la présente invention est d'améliorer encore la souplesse de traitement offerte par les mémoires TRIE, particulièrement dans les applications de routage.

[0016] L'invention propose ainsi un procédé pour associer des références d'acheminement à des paquets de données au moyen d'une mémoire TRIE, en analysant successivement, à partir de registres portiers différents de la mémoire TRIE, différentes portions d'un en-tête de chaque paquet contenant des informations de protocole. A l'arrivée d'un paquet, on stocke son en-tête dans une mémoire tampon et on analyse une première portion de l'en-tête stocké. Chaque analyse d'une portion de l'en-tête d'un paquet produit soit la référence d'acheminement associée au paquet, soit une référence intermédiaire comportant un premier code permettant de localiser, à un emplacement arbitraire de la mémoire tampon, une portion suivante à analyser et un second code permettant de localiser, à un emplacement arbitraire de la mémoire TRIE, un registre portier à partir duquel ladite portion suivante est à analyser. Après avoir analysé la première portion d'un en-tête stocké, on en analyse des portions suivantes conformément aux premiers et seconds codes contenus dans les références intermédiaires successivement produites, jusqu'à ce que la référence d'acheminement soit produite.

[0017] Ainsi, le contenu de la mémoire TRIE ne représente plus seulement les références proprement dites associées aux en-têtes de paquets. Elle intègre en outre un programme consistant en l'enchaînement des analyses élémentaires à effectuer en fonction des différentes configurations prises en compte par la mémoire. Ces enchaînements sont entièrement programmables, en ce que l'utilisateur peut définir arbitrairement, et à chaque étape du processus, quelle portion de l'en-tête doit être examinée et à partir de quel registre de la mémoire TRIE.

[0018] Un autre aspect de la présente invention se rapporte à un dispositif de traitement de paquets, comprenant un circuit d'analyse de l'en-tête des paquets reçus, à l'aide d'une mémoire associative de type TRIE fonctionnant conformément au procédé ci-dessus. Ce dispositif peut effectuer tout type de traitement nécessitant une analyse d'en-tête. Il peut notamment être un routeur IP, un dispositif de sécurité tel qu'un pare-feu (« firewall »), un dispositif de mesure, etc.

[0019] D'autres particularités et avantages de la présente invention apparaîtront dans la description ci-après d'exemples de réalisation non limitatifs, en référence aux dessins annexés, dans lesquels :

- la figure 1, précédemment commentée, montre un exemple de contenu d'une mémoire TRIE ;
- la figure 2, précédemment commentée, est un organigramme d'une procédure d'analyse classique exécutée pour consulter la mémoire TRIE ;

- la figure 3 est un schéma synoptique d'un routeur de paquets selon l'invention ;
- la figure 4 est un diagramme montrant la structure d'un exemple d'en-tête de paquet ;
- les figures 5 et 6 sont des diagrammes montrant la structure de cellules de la mémoire TRIE du routeur ; et
- la figure 7 est un organigramme d'une procédure d'analyse selon l'invention pour consulter une mémoire TRIE.

[0020] Pour illustrer la description ci-après, on considère le cas où les paquets acheminés par le routeur selon l'invention sont transportés sur un réseau à mode de transfert asynchrone (ATM), et on suppose que l'en-tête de chaque paquet est toujours contenu dans une cellule ATM.

[0021] Le routeur 10 représenté par la figure 3 fonctionne avec un ordinateur-hôte 11. L'ordinateur hôte 11 peut émettre et recevoir des paquets, notamment pour la gestion du processus de routage. Il dispose pour cela d'une voie virtuelle (VC) en entrée et en sortie du routeur 10.

[0022] Le routeur 10 comprend un module d'acheminement (« forwarding ») 12 qui achemine les paquets reçus selon des instructions, ci-après appelées « références d'acheminement » ou « status final », obtenues par un module d'analyse 13 à partir d'une mémoire 14 organisée comme un tableau de mémoire TRIE. Dans le cas d'un équipement de réseau ATM, le module d'acheminement 12 peut réaliser essentiellement une traduction des identifiants de conduits et de voies virtuels VPI/VCI (« Virtual Path Identifier / Virtual Channel Identifier »), la fusion des voies virtuelles selon les conduits virtuels, et la délivrance des paquets sur les ports de sortie du dispositif. Pour cela, il a besoin de connaître les couples VPI/VCI des paquets sortants, qui peuvent constituer les références d'acheminement stockées dans la mémoire TRIE 14.

[0023] Chaque cellule ATM contenant l'en-tête d'un paquet à router transite par une mémoire tampon 15 à laquelle le module d'analyse 13 a accès pour analyser des portions de ces en-têtes au moyen de la mémoire TRIE 14. Cette analyse est par exemple effectuée par quartets (K=4).

[0024] Configurer le routeur 10 consiste à enregistrer les données pertinentes dans la mémoire TRIE 14. Cette opération est réalisée par un module (non représenté) de gestion de la mémoire TRIE sous le contrôle de l'ordinateur-hôte 11. Les commandes de configuration peuvent être reçues dans des paquets transmis sur le réseau et destinés au routeur 10. Pour une façon de gérer dynamiquement le contenu de la mémoire TRIE 14, on pourra se reporter à la demande de brevet français 98 11856.

[0025] Dans l'exemple de routeur représenté sur la figure 3, le module d'analyse 13 coopère avec un automate 16 programmé pour effectuer certains contrôles et certaines actions sur les en-têtes des paquets, d'une manière dépendante des protocoles de communication supportés par le routeur. En dehors de cet automate 16, le fonctionnement du routeur 10 est indépendant des protocoles de transport des paquets.

[0026] La figure 4 montre un cas particulier de structure de l'en-tête d'un paquet pouvant être traité par le routeur selon l'invention. Dans cet exemple, le paquet est un paquet IPV4 (« Internet Protocol - Version 4 »), transportant le protocole applicatif UDP (« User Datagram Protocol »), encapsulé dans des cellules ATM par l'intermédiaire d'un protocole de niveau 2 de type LLC-SNAP (« Logical Link Control - Sub Network Access Protocol »). Les noms de champs indiqués sur la figure sont ceux employés dans les spécifications pertinentes concernant ces protocoles. L'en-tête global à analyser consiste en l'imbrication de l'en-tête de cinq octets de la cellule ATM, de l'en-tête LLC-SNAP, de l'en-tête du protocole IPV4 et de son extension UDP.

[0027] D'autres types de protocoles et d'encapsulations peuvent être supportés par le routeur 10. On peut par exemple prendre en compte le protocole PPP (« Point-to-Point Protocol ») au lieu du protocole SNAP au niveau 2, ou encore la technique MPLS (« Multi-Protocol Label Switching ») au niveau 2 et/ou au niveau 3. Dans ces conditions, les successions de type d'en-têtes suivantes sont susceptibles d'être rencontrées dans la trame AAL5 (« ATM Adaptation Layer n°5 ») suivant l'en-tête de cellule ATM :

```

IP
IP, IP
LLC-SNAP, IP
LLC-PPP, IP
MPLS
MPLS, ..., MPLS
LLC-SNAP, MPLS
MPLS, IP
MPLS, ..., MPLS, IP
MPLS, LLC-SNAP, IP
MPLS, ..., MPLS, LLC-SNAP, IP

```

IP désignant soit la version 4, soit la version 6 de l'Internet Protocol et les en-têtes des couches au-delà de la couche réseau (TCP, UDP...).

[0028] Dans le cas particulier LLC-SNAP/IPV4/UDP, la figure 4 montre en hachuré les champs de l'en-tête éventuellement pertinents pour l'opération de routage. Selon le type de routage effectué, certains de ces champs peuvent ne pas être pris en considération, en plus des champs non hachurés représentés sur la figure. Pour éviter une augmentation inutile de la taille de la mémoire TRIE, le contenu de ces champs est simplement ignoré au cours du processus d'analyse.

[0029] L'analyse porte ainsi sur des portions différentes de l'en-tête qui font successivement l'objet d'analyses partielles fournissant des références intermédiaires, jusqu'à la dernière portion pour laquelle la mémoire TRIE 14 fournit la référence d'acheminement destinée au module 12.

[0030] Le procédé selon l'invention permet d'analyser les champs de l'en-tête dans n'importe quel ordre. Ceci est utile dans un certain nombre de cas où l'analyse dans l'ordre d'apparition des champs s'avère insuffisante. Par exemple, on peut noter :

- qu'il est intéressant d'analyser le champ TOS (Type Of Service) de l'en-tête IP après l'adresse de destination, alors que ces champs apparaissent dans l'ordre inverse, afin de limiter la taille mémoire requise ;
- qu'avec des protocoles de réservation de type RSVP, il est nécessaire de pouvoir revenir en arrière sur les champs « Protocol » et « destination address » de l'en-tête IP après avoir analysé l'en-tête de niveau 4 ;
- que dans des applications de multidiffusion, il importe de pouvoir revenir en arrière sur le champ TTL (Time To Live) de l'en-tête IP ainsi que sur les identifiants VPI/VCI de l'entrée du routeur dans l'en-tête ATM afin de pouvoir empêcher les phénomènes de bouclage entre plusieurs noeuds du réseau...

[0031] L'organisation des données dans la mémoire TRIE 14, et la façon de les consulter, est adaptée afin de permettre la programmation de sauts quelconques entre les champs de l'en-tête analysés et entre les registres portiers de la mémoire TRIE.

[0032] Les figures 5 et 6 illustrent, dans un mode de réalisation particulier, la structure des données contenues dans les cellules élémentaires non vides de la mémoire TRIE 14. Dans cet exemple, chaque cellule élémentaire représente une zone de mémoire de 32 bits. Les trois premiers bits de la cellule de la mémoire TRIE forment un champ de commande CT destiné à contrôler l'état de l'automate 16.

[0033] A titre d'exemple, l'automate 16 peut avoir 5 états :

- « inactif », lorsqu'aucun en-tête n'est présent dans la mémoire tampon 15;
- « ATM », lorsque le module 13 est en train d'analyser les identifiants VPI/VCI de l'en-tête ATM ;
- « MPLS », lorsque le module 13 est en train d'analyser un en-tête MPLS ;
- « IP », lorsque le module 13 est en train d'analyser un en-tête IP ou ses extensions ;
- « autre », lorsque le module 13 est en train d'analyser un autre type d'en-tête (LLC, SNAP, PPP, ...).

[0034] Le codage du champ de commande CT des cellules élémentaires de la mémoire TRIE est par exemple le suivant :

- CT=000 : état de l'automate inchangé,
- CT=001 : fin d'analyse,
- CT=01a : transition vers l'état « autre »,
- CT=10a : transition vers l'état « IP »,
- CT=11a : transition vers l'état « MPLS ».

[0035] Le bit a ci-dessus indique si la trame provient de l'ordinateur-hôte 11 ou non afin que l'automate 16 inhibe le cas échéant le traitement des champs TTL et « checksum » de l'en-tête IPV4.

[0036] Dans l'exemple des figures 5 et 6, les dix-huit bits de poids les plus faibles de la cellule élémentaire de la mémoire TRIE forment un champ Ptr contenant soit un pointeur vers le prochain registre de la mémoire TRIE à partir duquel l'analyse est à poursuivre (registre portier ou non) dans le cas de la figure 5, soit la référence d'acheminement destinée au module 12 dans le cas de la figure 6.

[0037] Dans ce dernier cas, le contenu du champ de commande CT est 001, étant donné que la délivrance de la référence d'acheminement termine le processus d'analyse à l'aide de la mémoire TRIE.

[0038] Dans le cas de la figure 5, le pointeur contenu dans le champ Ptr peut désigner le registre de poursuite d'analyse directement ou indirectement selon le contenu d'un champ de comptage CP de deux bits. Un compteur peut être alloué à toute cellule de la mémoire TRIE. Ce compteur est alors incrémenté chaque fois que cette cellule est rencontrée dans le parcours effectué au cours de l'analyse. Chaque compteur V(A) est matérialisé dans un emplacement de mémoire couplé à un autre emplacement contenant un pointeur PT(A) vers un registre de poursuite d'analyse dans la mémoire TRIE.

[0039] Lorsque le contenu du champ de comptage CP est différent de 00, le champ Ptr de la cellule de la mémoire TRIE contient l'adresse de l'emplacement de mémoire où est mémorisé le compteur pertinent V(A), et l'emplacement PT(A) couplé à ce dernier contient le pointeur vers le registre de la mémoire TRIE à utiliser pour poursuivre l'analyse. Lorsque les deux bits du champ CP sont 00, le champ Ptr de la cellule élémentaire pointe directement vers le registre de poursuite d'analyse dans la mémoire TRIE.

[0040] Pour désigner quelle portion de l'en-tête du paquet doit faire l'objet de la suite de l'analyse, la cellule élémentaire ne contenant pas de référence d'acheminement (figure 5) comprend un champ de format FM de deux bits et un champ de déplacement DP de sept bits. Le champ FM indique le format de représentation numérique de l'emplacement visé de la mémoire tampon 15. Cette représentation peut être :

- séquentielle (FM=00) si les quartets à analyser se succèdent immédiatement dans l'en-tête stocké dans la mémoire 15,
- absolue (FM=11) pour désigner la position absolue du quartet indépendamment du sous-en-tête en cours d'analyse (ceci est utile par exemple pour retourner sur le champ VC de l'en-tête ATM au cours du traitement de l'en-tête IP dans des applications de multidiffusion),
- différentielle (FM=10), pour exprimer la position du prochain quartet par rapport au quartet courant, ce qui est utile lorsque la longueur de certains en-têtes n'est pas connue a priori (par exemple dans l'état « autre »),
- relative (FM=01) pour exprimer la position du prochain quartet par rapport à un emplacement déterminé de la mémoire tampon 15, repéré par une valeur de décalage gérée par l'automate 16. Cette valeur de décalage OFFSET désigne typiquement le début du sous-en-tête en cours d'analyse. L'automate 16 le gère en ajoutant lors des transitions d'état la longueur de l'en-tête correspondant à l'état antérieur.

[0041] Les autres tâches assurées par l'automate 16 comprennent les vérifications d'erreurs effectuées sur la base du champ « checksum » de l'en-tête IPV4 et les manipulations éventuelles des champs TOS et TTL de l'en-tête IP. Ces modifications éventuelles sont effectuées à l'issue du processus d'analyse, sur la base de paramètres fournis par la mémoire TRIE avec la référence d'acheminement. Ces paramètres sont contenus dans un champ TT de deux bits et dans un champ PA de huit bits de chaque cellule élémentaire contenant une référence d'acheminement (figure 6). Le codage est par exemple le suivant :

- TT=00 : aucune modification des champs TTL et TOS (utilisable pour les paquets destinés à l'ordinateur-hôte 11) ;
- TT=01 : décrémenter TTL de 1 ;
- TT=10 : décrémenter TTL du contenu du champ PA ;
- TT=11 : décrémenter TTL de 1 et remplacer TOS par le contenu du champ PA.

[0042] Au moment d'une transition de l'état « ATM » ou « MPLS » ou « autre » vers l'état « IP » en version 4, l'automate 16 vérifie que le contenu du champ « checksum » de l'en-tête IPV4 est conforme au reste de l'en-tête selon le codage détecteur d'erreurs appliqué dans ces en-têtes conformément aux spécifications. Au moment de la transition de l'état « IP » vers l'état « inactif » (CT=001), l'automate 16 met à jour le champ « checksum » de l'en-tête IPV4 en faisant la somme de celui contenu dans l'en-tête entrant et du code détecteur d'erreur calculé sur la base des modifications faites dans les champs TTL et/ou TOS.

[0043] La figure 7 montre l'organigramme d'une procédure d'analyse que peut mettre en oeuvre le module 13 dans les exemples considérés ci-dessus. A l'initialisation 20, le numéro i du quartet analysé est mis à 0 pour désigner le début de l'en-tête ATM dans la mémoire tampon 15, le registre portier de base R_0 est sélectionné comme registre R, et l'automate 16 reçoit une commande d'activation (passage de l'état « inactif » à l'état « ATM »).

[0044] Dans chaque itération i, le contenu C de la cellule élémentaire $T[R, V_i]$ du registre sélectionné R, désignée par le (i+1)-ième quartet V_i de l'en-tête stocké dans la mémoire tampon 15, est lu à l'étape 21. Si le champ de commande CT(C) de cette cellule est différent de 001 (test 22), le contenu de ce champ de commande est fourni à l'automate 16 à l'étape 23 pour qu'il adopte le cas échéant les changements d'état requis en procédant aux actions correspondantes, puis le module 13 détermine comment l'analyse doit être poursuivie, en identifiant la position dans la mémoire tampon 15 du prochain quartet à analyser à l'étape 24, ainsi que le prochain registre de la mémoire TRIE à utiliser pour poursuivre l'analyse aux étapes 25 à 28.

[0045] A l'étape 24, l'index du prochain quartet à analyser est calculé en fonction de l'index du quartet courant i, du code de position représenté par les champs FM et DP de la cellule de la mémoire TRIE et éventuellement par le paramètre de décalage OFFSET géré par l'automate 16. Avec les conventions précédemment considérées, l'expression de la fonction f est :

$$f(i, 00, DP, OFFSET) = i + 1 ;$$

$$f(i,11,DP,OFFSET) = DP ;$$

$$f(i,10,DP,OFFSET) = i + DP, \text{ avec } DP \text{ positif ou négatif ; et}$$

$$f(i,01,DP,OFFSET) = OFFSET + DP, \text{ avec } DP \text{ positif ou négatif.}$$

[0046] A l'étape 25, le module 13 examine le champ CP de la cellule courante de la mémoire TRIE pour déterminer si un comptage est nécessaire. Si aucun comptage n'est nécessaire (CP=00), le registre désigné par le pointeur lu dans le champ Ptr de la cellule est sélectionné comme registre R pour l'itération suivante à l'étape 26. Si CP(C) ≠ 00 à l'étape 25, l'adresse A du compteur à incrémenter est obtenue à l'étape 27 comme étant le contenu du champ Ptr de la cellule courante de la mémoire TRIE. Ensuite, à l'étape 28, le module 13 commande l'incrément de ce compteur V(A) et récupère le pointeur PT(A) afin de sélectionner le registre R pour l'itération suivante.

[0047] Ce processus d'analyse se poursuit en enchaînant plusieurs analyses logiques fournissant des status intermédiaires. Il se termine lorsque la valeur 001 apparaît dans le champ de commande CT(C) de la cellule courante (test 22). A ce moment, le module 13 fournit à l'automate 16 les paramètres contenus dans les champs TT et PA de la cellule à l'étape 29, puis fournit le status final Ptr(C) au module d'acheminement 12 à l'étape 30.

Revendications

1. Procédé pour associer des références d'acheminement à des paquets de données au moyen d'une mémoire TRIE (14), en analysant successivement, à partir de registres portiers différents de la mémoire TRIE, différentes portions d'un en-tête de chaque paquet contenant des informations de protocole, caractérisé en ce qu'à l'arrivée d'un paquet, on stocke son en-tête dans une mémoire tampon (15) et on analyse une première portion de l'en-tête stocké, en ce que chaque analyse d'une portion de l'en-tête d'un paquet produit soit la référence d'acheminement associée au paquet, soit une référence intermédiaire comportant un premier code (FM, DP) permettant de localiser, à un emplacement arbitraire de la mémoire tampon, une portion suivante à analyser et un second code (Ptr) permettant de localiser, à un emplacement arbitraire de la mémoire TRIE, un registre portier à partir duquel ladite portion suivante est à analyser, et en ce qu'après avoir analysé la première portion d'un en-tête stocké, on en analyse des portions suivantes conformément aux premiers et seconds codes contenus dans les références intermédiaires successivement produites, jusqu'à ce que la référence d'acheminement soit produite.
2. Procédé selon la revendication 1, dans lequel la mémoire TRIE (14) consiste en des cellules élémentaires ayant chacune un champ (CT) indiquant si la cellule contient une référence d'acheminement, et dans lequel chaque cellule élémentaire ne contenant pas de référence d'acheminement a au moins un premier champ (FM, DP) pour contenir une information désignant un emplacement de la mémoire tampon (15) où l'analyse est à poursuivre et un second champ (Ptr) pour contenir une information désignant un registre de poursuite d'analyse de la mémoire TRIE.
3. Procédé selon la revendication 2, dans lequel ladite information désignant un emplacement de la mémoire tampon (15) où l'analyse est à poursuivre comporte un code de format (FM) indiquant la représentation numérique dudit emplacement de la mémoire tampon et un code d'adresse (DP) désignant ledit emplacement de la mémoire tampon selon ladite représentation.
4. Procédé selon la revendication 3, dans lequel le code de format (FM) indique (i) une représentation séquentielle, où les éléments de données de l'en-tête stocké sont analysés dans leur ordre d'apparition, (ii) une représentation absolue, où le prochain élément de données à analyser est désigné par sa position absolue dans l'en-tête stocké, (iii) une représentation différentielle, où le prochain élément de données à analyser est désigné par sa position par rapport à l'élément de données en cours d'analyse, ou (iv) une représentation relative, où le prochain élément de données à analyser est désigné par sa position par rapport à une portion déterminée de l'en-tête stocké.
5. Procédé selon l'une quelconque des revendications 2 à 4, dans lequel chaque cellule élémentaire ne contenant pas de référence d'acheminement a un champ supplémentaire (CC) pour contenir une information indiquant si une opération de comptage est requise, et dans lequel chaque cellule élémentaire ne contenant pas de référence d'acheminement et dont ledit champ supplémentaire (CC) indique qu'une opération de comptage est requise a

EP 1 030 493 A1

dans son second champ (Ptr) un pointeur (A) vers un emplacement de mémoire (V(A)) contenant un compteur à incrémenter lors de ladite opération et associé à un autre emplacement de mémoire (PT(A)) contenant un pointeur (R) vers le registre de poursuite d'analyse de la mémoire TRIE (14).

- 5 6. Procédé selon l'une quelconque des revendications 1 à 5, dans lequel chaque cellule élémentaire de la mémoire TRIE (14) comporte un champ de commande (CT) pour contenir des instructions destinées à un automate auxiliaire (16).
- 10 7. Procédé selon la revendication 6, dans lequel une référence d'acheminement est délivrée depuis la mémoire TRIE (14) avec des paramètres fournis à l'automate (16) pour spécifier des modifications éventuelles à opérer sur une portion de l'en-tête du paquet stocké dans la mémoire tampon (15).
- 15 8. Dispositif de traitement de paquets (10), comprenant un circuit (13) d'analyse de l'en-tête de paquets reçus, à l'aide d'une mémoire associative de type TRIE (14) fonctionnant selon un procédé conforme à l'une quelconque des revendications précédentes.

20

25

30

35

40

45

50

55

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1					1		4	7								
2						2					3					
3												<u>S1</u>				
4					<u>S0</u>			5			6					
5												<u>S2</u>				
6																
7									8							
8									9							
9											<u>S3</u>	10				
10														<u>S0</u>		
...																

FIG. 1

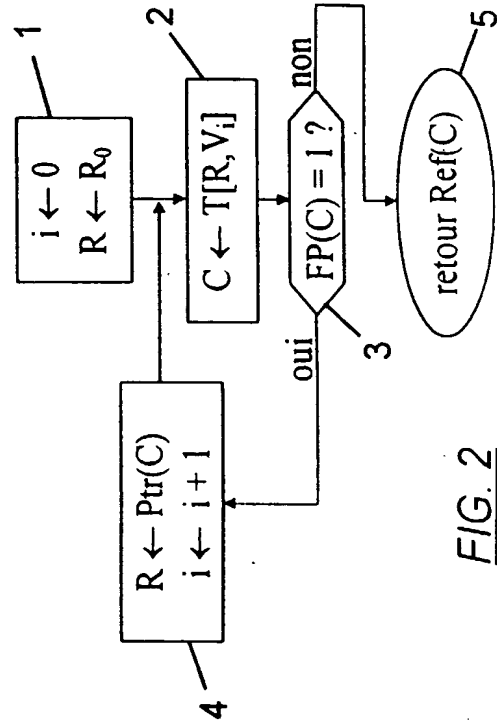


FIG. 2

FIG.3

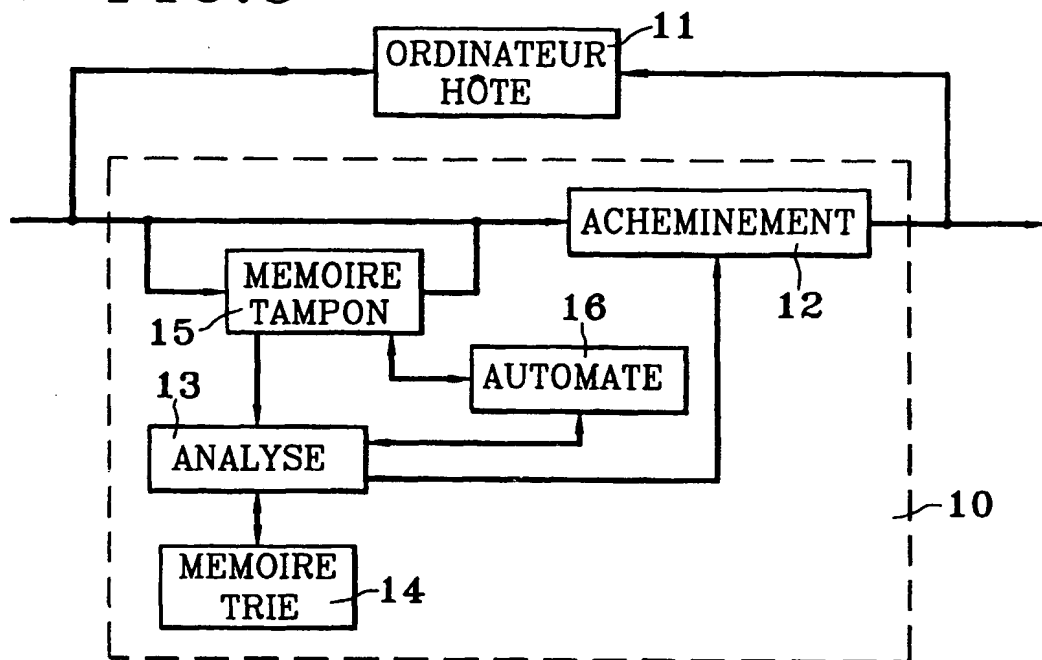


FIG.5

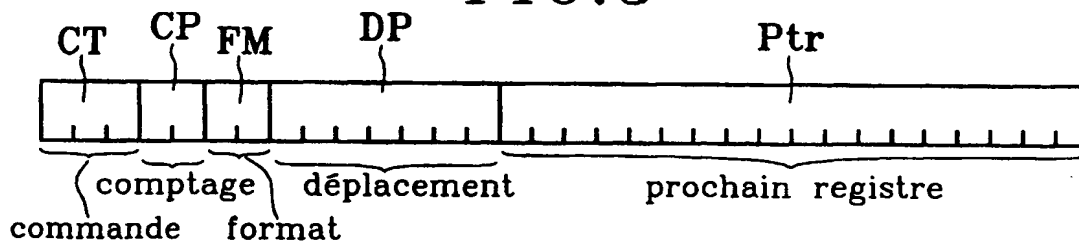


FIG.6

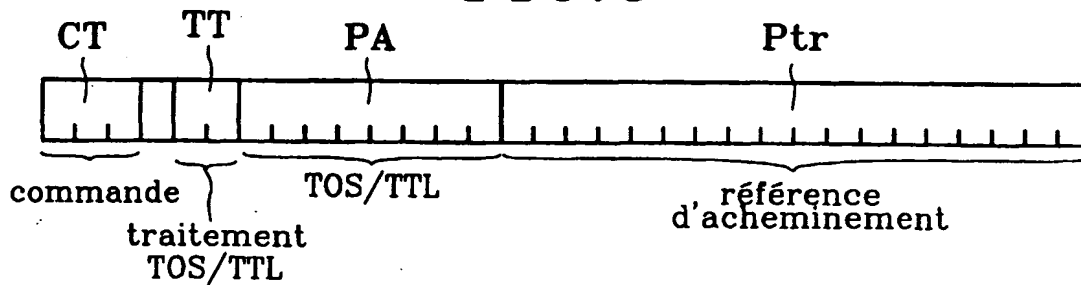


FIG. 4

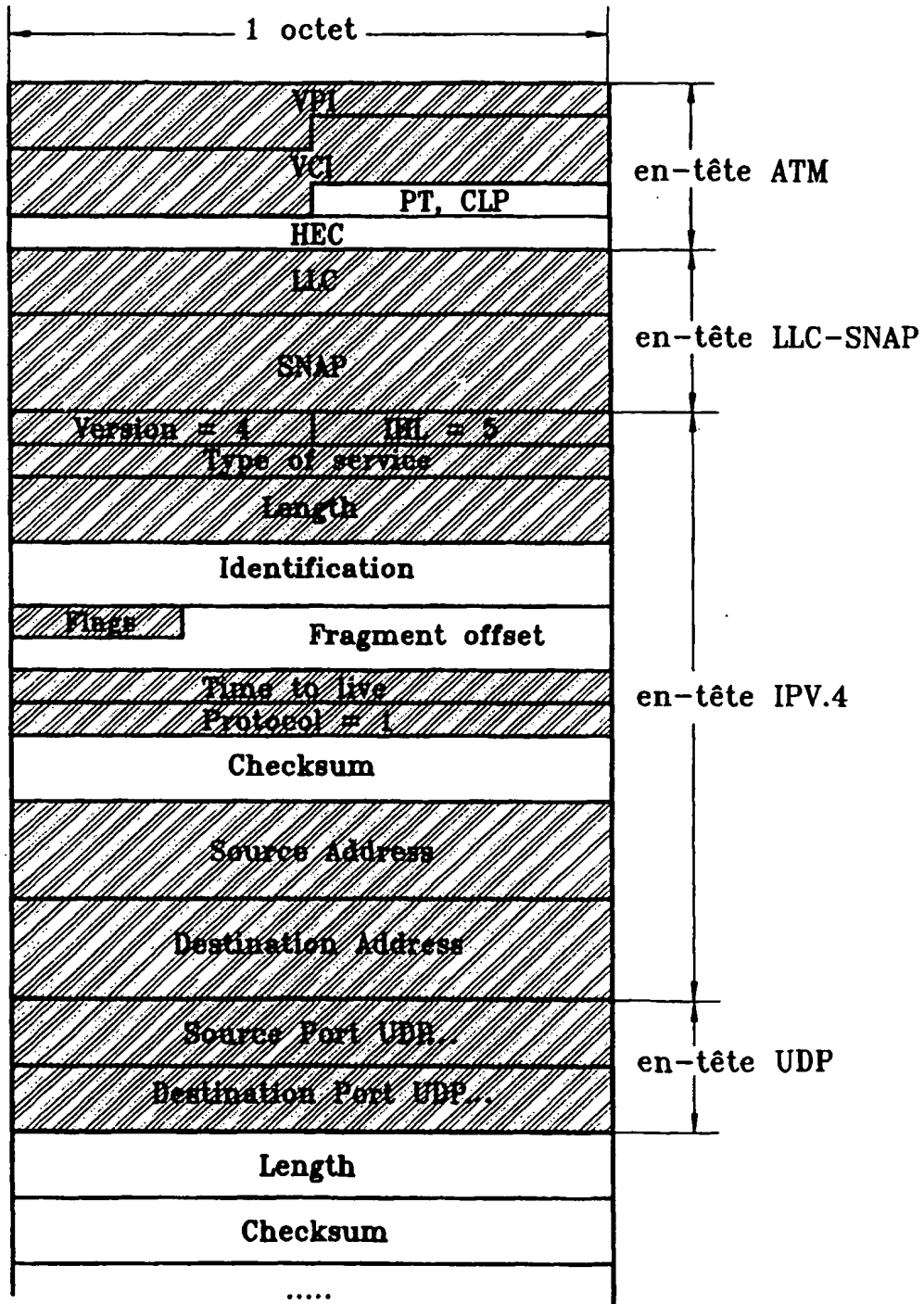
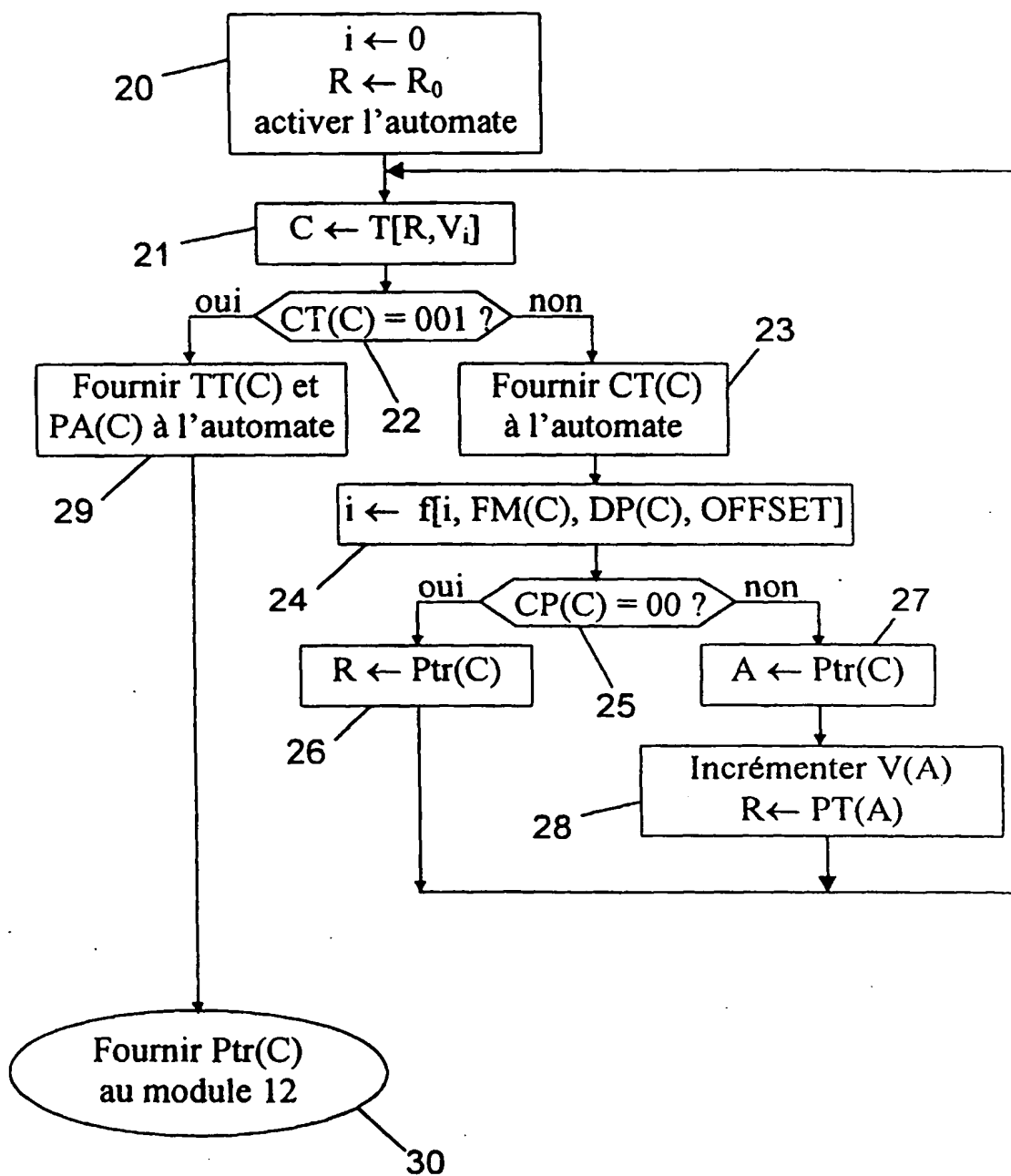


FIG. 7





Office européen
des brevets

RAPPORT DE RECHERCHE EUROPEENNE

Numéro de la demande
EP 00 40 0366

DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int.Cl.7)
X	WO 98 27662 A (JUNIPER NETWORKS) 25 juin 1998 (1998-06-25) * abrégé * * page 1, ligne 1 - page 9, ligne 5 *	1-8	H04L29/06
X	WO 96 00945 A (IBM ;DOERINGER WILLIBALD (CH); DYKEMAN DOUGLAS (CH); KARJOTH GUENT) 11 janvier 1996 (1996-01-11) * abrégé * * page 1, ligne 1 - page 10, ligne 11 *	1-8	
X	SRINIVASAN V ET AL: "Fast and scalable layer four switching" ACM SIGCOMM'98 CONFERENCE. APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATION, VANCOUVER, BC, CANADA, 2-4 SEPT. 1998, vol. 28, no. 4, pages 191-202, XP002107832 ISSN 0146-4833, Computer Communication Review, Oct. 1998, ACM, USA * le document en entier *	1-8	
D,X	FR 2 707 775 A (PELAMOURGUES LIONEL;DURET CHRSTIAN) 20 janvier 1995 (1995-01-20) * le document en entier *	1-8	DOMAINES TECHNIQUES RECHERCHES (Int.Cl.7) H04L
D,X	TONG-BI PEI ET AL: "PUTTING ROUTING TABLES IN SILICON" IEEE NETWORK: THE MAGAZINE OF COMPUTER COMMUNICATIONS, vol. 6, no. 1, 1 janvier 1992 (1992-01-01), pages 42-50, XP000262202 * le document en entier *	1-8	
Le présent rapport a été établi pour toutes les revendications			
Lieu de la recherche LA HAYE		Date d'achèvement de la recherche 2 juin 2000	Examineur Katerbau, R
CATEGORIE DES DOCUMENTS CITES X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	

EPO FORM 1503 03.82 (P4/C22)



Office européen
des brevets

RAPPORT DE RECHERCHE EUROPEENNE

Numéro de la demande
EP 00 40 0366

DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int.CI.7)
A	EP 0 380 294 A (CODEX CORP) 1 août 1990 (1990-08-01) * abrégé * * page 2, colonne 1, ligne 1 - page 3, colonne 1, ligne 55 * -----	1-8	
			DOMAINES TECHNIQUES RECHERCHES (Int.CI.7)
Le présent rapport a été établi pour toutes les revendications			
Lieu de la recherche LA HAYE		Date d'achèvement de la recherche 2 juin 2000	Examineur Katerbau, R
<p>CATEGORIE DES DOCUMENTS CITES</p> <p>X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire</p> <p>T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant</p>			

EPO FORM 1503 03 82 (P04/03/02)

**ANNEXE AU RAPPORT DE RECHERCHE EUROPEENNE
RELATIF A LA DEMANDE DE BREVET EUROPEEN NO.**

EP 00 40 0366

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche européenne visé ci-dessus.
Lesdits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets.

02-06-2000

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 9827662 A	25-06-1998	US 5909440 A	01-06-1999
		US 5905725 A	18-05-1999
		EP 0947078 A	06-10-1999
		EP 0948849 A	13-10-1999
		EP 0940025 A	08-09-1999
		WO 9827660 A	25-06-1998
		WO 9827697 A	25-06-1998
WO 9600945 A	11-01-1996	DE 69422935 D	09-03-2000
		EP 0804769 A	05-11-1997
		US 5787430 A	28-07-1998
FR 2707775 A	20-01-1995	DE 69414934 D	14-01-1999
		DE 69414934 T	01-07-1999
		EP 0639013 A	15-02-1995
		JP 7095197 A	07-04-1995
		US 5781431 A	14-07-1998
EP 0380294 A	01-08-1990	AT 152260 T	15-05-1997
		AU 624205 B	04-06-1992
		AU 4704089 A	26-07-1990
		CA 2007168 A	23-07-1990
		DE 69030519 D	28-05-1997
		HK 1009317 A	28-05-1999
		JP 2233025 A	14-09-1990
		JP 2830280 B	02-12-1998
		US 5329405 A	12-07-1994

EPO FORM P0480

Pour tout renseignement concernant cette annexe : voir Journal Officiel de l'Office européen des brevets, No.12/82

A METHOD OF ASSOCIATING ROUTING REFERENCES WITH
DATA PACKETS BY MEANS OF A TRIE MEMORY AND A
PACKET ROUTER APPLYING THIS METHOD

5 The present invention relates to associative memories and in particular memories of the «TRIE» type (derived from the English verb «reTRIEve»).

10 The principle of the «TRIE» memory was proposed by R. de la Briandais and E. Fredkin et al towards the end of the 1950s (see E.Fredkin et al.: « *Trie Memory* », Communications of the ACM, Vol. 3, No. 9, September 1960, pages 490-499). It consists in cutting up the bit strings to be recognised into successive slices of a fixed length (of K bits) and integrating them in a two-
15 dimensional table T . Each row of the table constitutes a register of 2^K elementary cells. A register (R) is assigned to each slice of the string and a cell in the register is associated with the value (V), ranging between 0 and 2^K-1 of this slice. The contents
20 ($C=T[R,V]$) of the cell determined in this manner represent either the register allocated to the subsequent slice (or *pointer*) or an end of analysis reference (or « *status* ») if the analysis of the string must end on this slice.

25 The register allocated to the first slice of the string, which is also the point of entry to the table, is also referred to as the *carrier*. The data to be analysed in the form of bit strings, i.e. to be compared with the contents of the TRIE memory, will also be
30 referred to as *routes* hereafter. The term *path* will be used to denote the succession of concatenated cells in the table associated with a route. Each register of the table will be said to be of the order of i_0 if it is attributed to the $(i+1)$ -th slice of one or more stored

routes. The carrier register will therefore be in the order of 0. The TRIE memory associates with each of the registers in the order of i_0 a unique sequence of iK bits corresponding to the iK first bits of each route whose path in the table passes via a cell of the register in question.

The following example will provide an illustration of how data is stored in a TRIE memory in the specific case where $K=4$. The value of each slice is represented by a digit in hexadecimal numbering (0,1,...,E,F) and each of the registers contains $2^4=16$ cells.

Let us assume that the routes to be recognised are those commencing with the codes 45A4, 45AB, 67AB, 788A and 788BD, to which the statuses S0, S1, S2, S3 and S0 have been allocated respectively (a same status may be shared by several routes). By using the row index for the register R and the column index for the value V of the slices and by taking the register $R_0=0$ as the carrier, the table of the TRIE memory will appear as illustrated in figure 1, the underlined data being the statuses. The codes 45A4, 45AB, 67AB, 788A and 788BD are represented respectively in the table of figure 1 by the paths:

$T[0, \underline{4}] _ T[1, \underline{5}] _ T[2, \underline{A}] _ T[3, \underline{4}] \ ;$
 $T[0, \underline{4}] _ T[1, \underline{5}] _ T[2, \underline{A}] _ T[3, \underline{B}] \ ;$
 $T[0, \underline{6}] _ T[4, \underline{7}] _ T[5, \underline{A}] _ T[6, \underline{B}] \ ;$
 $T[0, \underline{7}] _ T[7, \underline{8}] _ T[8, \underline{8}] _ T[9, \underline{A}] \ ;$
 $T[0, \underline{7}] _ T[7, \underline{8}] _ T[8, \underline{8}] _ T[9, \underline{B}] \ ; _ T[10, \underline{D}] \ .$

30

From this example, it may be seen that all the codes starting with a common part of iK bits are represented by a common initial path in the memory

leading to the register of order i with which the sequence formed by these iK bits is associated.

If we consider a route to be analysed, cut up into a series of binary slices of values V_i where $0 \leq i \leq N$ and
 5 $\{R_i\}$ is the series of registers associated with the values V_i , where R_0 still denotes the carrier register, the analysis algorithm implemented may be that illustrated in figure 2.

On initialisation 1 of this algorithm, the analysis
 10 row i is set to 0 and the carrier register R_0 is selected as the register R . In each iteration of row i , the contents C of the cell $T[R, V_i]$, denoted by the $(i+1)$ -th slice V_i of the route in the register of order i selected, is read at step 2. If this cell contains a
 15 continue analysis pointer, which will indicate at test 3 the value 1 for a bit $FP(C)$ stored in the cell, the register of order $i+1$ denoted by this pointer $Ptr(C)$ is selected as the register R for the next iteration at step 4 and the row i is incremented. If test 3 reveals a
 20 cell which does not contain a pointer ($FP(C)=0$), the status $Ref(C)$ read in the cell concerned is returned at step 5 as a result of looking up the table.

This algorithm enables routes containing any number of slices to be analysed. A same table may be used for
 25 several types of analysis, managing data on the basis of different carriers. Furthermore, it enables the analysis time of the data to be controlled: analysing a number N of slices of K bits will require at most N times the duration of one iteration.

30 The algorithm of figure 2 may be implemented very rapidly by a hardware component managing accesses to the table memory. In particular, it will enable high-performance routers to be set up for packet-switched telecommunications networks. The header of the packets

is analysed by the component on the fly and the status associated with a route designates, for example, an output port of the router to which the packets bearing a destination address conforming to this route must be
5 routed.

Such a router may be a multi-protocol router. This being the case, the different sections of the header are analysed on the basis of different carriers. For example, a first analysis of a header field (or several)
10 indicating the protocol used and/or the version of this protocol may be analysed from a first carrier. This first analysis will provide a reference which, although corresponding to a logical end of the analysis, may be incorporated in the TRIE memory by a continue analysis
15 pointer denoting another carrier register to be used for analysing the rest of the header. The reference in question may also trigger time delays or skips by a given number of bits in the header being analysed in order to be able to choose which portion of the header
20 should be analysed next. In practice, a certain number of analyses are generally run in succession in order to trigger the operations required by the protocols supported, depending on the content of the headers. One of these analyses will relate to the destination address
25 needed to complete the routing function strictly speaking.

A router of the type outlined above is described in French patent 2 707 775. On the subject of using a TRIE memory in routers, reference may be made to the article
30 « *Putting Routing Tables in Silicon* » by T.B. Pei et al., IEEE Network Magazine, January 1992, pages 42-50.

The fact of being able to string together several elementary analyses and insert programmable skips between them lends a high degree of flexibility to the
35 method, particularly when processing encapsulated

protocols of the ISO model in several layers. Analysing slices of the header on the fly as they arrive also enhances speed.

However, in a certain number of situations, it is
5 useful to be able to return to the header in order to examine certain fields in an order other than that in which they arrived. This will often allow better optimisation of the memory size required. It is also a feature required by certain protocols, such as the RSVP
10 reservation protocol or the multicast protocols.

One objective of the present invention is to improve the processing flexibility offered by TRIE memories still further, especially in routing applications.

15 Accordingly, the invention proposes a method of associating routing references with data packets by means of a TRIE memory, whereby different carrier registers of the TRIE memory analyse in succession different portions of a header of each packet containing
20 protocol data. When a packet arrives, its header is stored in a buffer memory and a first portion of the stored header is analysed. Each analysis of a header portion of a packet produces either the routing reference associated with the packet or an intermediate
25 reference containing a first code, which makes it possible to locate at an arbitrary location of the buffer memory a subsequent portion to be analysed, and a second code, which makes it possible to locate at an arbitrary location of the TRIE memory a carrier register
30 from which said subsequent portion should be analysed. Having analysed the first portion of a stored header, the subsequent portions thereof are analysed in accordance with the first and second codes contained in the intermediate references produced in succession until
35 the routing reference is obtained.

Consequently, the contents of the TRIE memory no longer represent only the reference associated with the packet headers as such. They also incorporate a programme consisting of the string of elementary
 5 analyses to be performed, depending on the different configurations taken into account by the memory. These strings are entirely programmable insofar as the user can define, arbitrarily and at each step of the process, which portion of the header must be examined and from
 10 which register of the TRIE memory.

Another aspect of the present invention relates to a packet router having a circuit for analysing the header of the packets received using an associative memory of the TRIE type which operates in accordance
 15 with the method outlined above.

Other features and advantages of the present invention will become clearer from the description of examples given below, which is not restrictive in any respect, and with reference to the appended drawings, in
 20 which:

- figure 1, discussed above, shows an example of the contents of a TRIE memory;
- figure 2, discussed above, is a flow chart illustrating a conventional analysis method run as
 25 a means of looking up the TRIE memory;
- figure 3 is a synoptic diagram of a packet router as proposed by the invention;
- figure 4 is a diagram showing the structure of an example of a packet header;
- figure 5 and 6 are diagrams showing the structure of cells in the TRIE memory of the router; and
 30
- figure 7 is a flow chart illustrating an analysis procedure as proposed by the invention in order to look up a TRIE memory.

35 As a means of illustrating the description below,

we will look at a situation in which the packets routed by the router proposed by the invention are transported on an asynchronous transfer mode (ATM) network and it will be assumed that the header of each packet is always
5 contained in an ATM cell.

The router 10 illustrated in figure 3 operates in conjunction with a host computer 11. The host computer 11 can transmit and receive packets, in particular as a means of managing the routing process. To this end, it
10 has a virtual channel (VC) at the input and output of the router 10.

The router 10 has a routing («forwarding») module 12 which forwards the packets received in accordance with instructions, referred to hereafter as «forwarding
15 references» or «final status», produced by an analysis module 13 from an organised memory 14 such as a TRIE memory table. In the case of equipment used with an ATM network, the forwarding module 12 is essentially capable of translating the virtual path and virtual channel
20 identifiers VPI/VCI («Virtual Path Identifier / Virtual Channel Identifier»), merging the virtual channels depending on the virtual paths and applying packets to the output ports of the device. To this end, it needs to know the VPI/VCI pairs of outgoing packets, which may
25 constitute the forwarding references stored in the TRIE memory 14.

Each ATM cell containing the header of the packet to be routed passes via a buffer memory 15, to which the analysis module 13 has access in order to analyse
30 portions of these headers by means of the TRIE Memory 14. This analysis is performed by quartets ($K=4$), for example.

Configuring the router 10 consists in storing the relevant data in the TRIE memory 14. This operation is
35 performed by a module (not illustrated) which manages

the TRIE memory under the control of the host computer 11. The configuration commands may be received in packets transmitted across the network and addressed to the router 10. For details on how the contents of the TRIE memory 14 can be dynamically managed, reference 5 should be made to French patent application 98 11856.

In the case of the router illustrated in figure 3, the analysis module 13 co-operates with a controller 16 programmed to run certain checks and apply actions to 10 the packet headers independently of the communication protocols supported by the router. Outside of this controller 16, operation of the router 10 is independent of the carrier protocols of the packets.

Figure 4 illustrates a specific example of a header 15 structure for a packet that can be processed by the router of the invention. In this example, the packet is an IPV4 packet («Internet Protocol-Version 4») carrying the UDP application protocol («User Datagram Protocol»), encapsulated in the ATM cells by means of a level 2 20 protocol of the LLC-SNAP type («Logical Link Control - Sub Network Access Protocol»). The field names indicated in the drawing are those used in the relevant specifications pertaining to these protocols. The global header to be analysed consists of an interleaved 25 arrangement of the five octet header of the ATM cell, the LLC-SNAP header, the header of the IPV4 protocol and its UDP extension.

Other types of protocol and encapsulation may be supported by the router 10. For example, it is also 30 possible to handle the PPP protocol («Point-to-Point Protocol») instead of the SNAP protocol at level 2 or alternatively MPLS technology («Multi-Protocol Label Switching») at level 2 and/or level 3. Under these conditions, the following are the series of header types 35 likely to be encountered in the AAL5 frame («ATM

Adaptation Layer № 5») following the header of the ATM cell:

```

      IP
      IP, IP
5     LLC-SNAP, IP
      LLC-PPP, IP
      MPLS
      MPLS,...MPLS
      LLC-SNAP, MPLS
10    MPLS, IP
      MPLS,...MPLS, IP
      MPLS, LLC-SNAP, IP
      MPLS,...,MPLS, LLC-SNAP, IP

```

where IP denotes either version 4 or version 6 of the Internet Protocol and the headers of layers beyond the network layer (TCP, UDP...).

In the specific case of LLC-SNAP/IPV4/UDP, the hashed lines of figure 4 show the header fields which may be relevant to the routing operation. Depending on the type of routing applied, some of these fields may not be taken into account, along with the non-hashed fields illustrated in the drawing. In order to avoid increasing the size of the TRIE memory unnecessarily, the content of these fields is simply ignored during the analysis process.

The analysis will therefore focus on the different portions of the header which will be partially analysed in succession to supply intermediate references until the final portion is reached for which the TRIE memory supplies the routing reference destined for the module 12.

The method proposed by the invention allows the header fields to be analysed in any order. This is useful in a certain number of situations where it is not

sufficient to analyse the fields in the order in which they appear. It should be pointed out, for example, that:

- 5 - it is of interest to analyse the TOS field (Type of Service) of the IP header after the destination address although these fields appear in the reverse order, so as to keep down the size of memory required;
- 10 - with reservation protocols of the RSVP type, it is necessary to be able to go back to the «Protocol» and «destination address» fields of the IP header after analysing the level 4 header;
- 15 - in multicast applications, it is necessary to be able to go back to the TTL (Time to Live) field of the IP header and to the VPI/VCI identifiers of the router input in the ATM header so as to prevent the occurrence of looping between several network nodes...

20 Organisation of the data in the TRIE memory 14 and the way in which they are looked up is adapted to make it possible to programme any skips that might be needed between the header fields being analysed and between the carrier registers of the TRIE memory.

25 In the case of one specific embodiment, figures 5 and 6 illustrate the structure of data contained in the non-empty elementary cells of the TRIE Memory 14. In this example, each elementary cell represents a 32 bit memory zone. The first three bits of the cell in the TRIE memory form a command field CT intended to control
30 the status of the controller 16.

By way of example, the controller 16 may have 5 statuses:

- «inactive» if no header is present in the buffer memory 15;
- 35 - «ATM» if the module 13 is in the process of

analysing the VPI/VCI identifiers of the ATM header;

- «MPLS» if the module 13 is in the process of analysing a MPLS header;
- 5 - «IP» if the module 13 is in the process of analysing an IP header or its extensions;
- «other» if the module 13 is in the process of analysing another type of header (LLC, SNAP, PPP,...).

10 The command field CT of the elementary cells of the TRIE memory will be coded, for example, as follows:

- CT=000: controller status unchanged,
- CT=001: end of analysis,
- CT=01a: transition to the «other» status,
- 15 - CT=10a: transition to the «IP» status,
- CT=11a: transition to the «MPLS» status.

20 The bit a above indicates whether or not the frame has come from the host computer 11 so that the controller 16 can inhibit processing of the TTL and «checksum» fields of the IPV4 header if necessary.

25 In the examples illustrated in figures 5 and 6, the smallest eighteen weighting bits of the elementary cell of the TRIE memory form a field Ptr containing either a pointer to the next register of the TRIE memory from which analysis should be continued (carrier register or not) in the case of figure 5 or the forwarding reference destined for the module 12 in the example illustrated in figure 6.

30 In the latter case, the command field CT contains 001 in view of the fact that issuing the forwarding reference will terminate the analysis process using the TRIE memory.

35 In the situation illustrated in figure 5, the pointer contained in the field Ptr may indicate directly or indirectly the register in which analysis should be

continued depending on what is contained in a two bit counting field CP. A counter may be allocated to every cell of the TRIE memory. This counter will then be incremented every time this cell is encountered in the route followed during the analysis. Each counter V(A) is placed in a memory location coupled with another location containing a pointer PT(A) to a register in the TRIE memory at which the analysis will be continued.

If the counting field CP contains something other than 00, the field Ptr of the TRIE memory cell contains the address of the memory location at which the relevant counter V(A) is stored and the location PT(A) coupled with this latter contains the pointer to the register in the TRIE memory which must be used to continue the analysis. If the two bits of the field CP are 00, the field Ptr of the elementary cell will point directly to the register in the TRIE memory at which analysis should be continued.

In order to indicate which portion of the packet header should be analysed next, the elementary cell which does not contain a forwarding reference (figure 5) will include a two bit format field FM and a seven bit displacement field DP. The field FM indicates the format of the numerical representation of the relevant location in the buffer memory 15.

This representation may be of the following type:

- sequential (FM=00) if the quartets to be analysed follow immediately one after the other in the header stored in the memory 15,
- absolute (FM=11) to denote the absolute position of the quartet independently of the sub-header being analysed (this is useful, for example, if it is necessary to return to the VC field of the ATM header when processing the IP header in multicast applications),

- differential (FM=10) in order to express the position of the next quartet relative to the current quartet, which is useful if the length of certain headers is not known a priori (for example in the «other» status),
- relative (FM=01) in order to express the position of the next quartet relative to a given location of the buffer memory 15, located by an offset value managed by the controller 16. This OFFSET value typically denotes the start of the sub-header currently being analysed. The controller 16 manages it by adding, during status changeovers, the length of the corresponding header in the previous status.

The other tasks carried out by the controller 16 include the error checksums based on the «checksum» field of the IPV4 header and any manipulations in the TOS and TTL fields of the IP header. Any such modifications are applied at the end of the analysis process on the basis of the parameters supplied by the TRIE memory with the forwarding reference. These parameters are contained in a two bit field TT and in an eight bit field PA of each elementary cell containing a forwarding reference (figure 6). The coding will be, for example, as follows:

- TT=00: no modification to the TTL and TOS fields (can be used for packets destined for the host computer 11);
- TT=01: decrement TTL by 1;
- TT=10: decrement TTL by the contents of the field PA;
- TT=11: decrement TTL by 1 and replace TOS with the contents of the field PA.

At the instant of changeover from «ATM» or «MPLS» or «other» status to «IP» status in version 4, the controller 16 checks to ensure that the contents of the

«checksum» field of the IPV4 header are consistent with the remainder of the header depending on the error detection coding applied in these headers in accordance with the specifications. At the instant of transition
 5 from «IP» status to «inactive» status (CT=001), the controller 16 updates the «checksum» field of the IPV4 header by summing that contained in the incoming header and the error detection code calculated on the basis of the modifications applied to the TTL and/or TOS fields.

10 Figure 7 gives a flow chart illustrating an analysis procedure that can be run by the module 13 in the examples described above. On initialisation 20, number i of the quartet analysed is set to 0 to indicate the start of the ATM header in the buffer memory 15, the base carrier register R_0 is selected as the register R and the controller 16 receives an activate command (switch from «inactive» status to «ATM» status).

In each iteration i , the contents C of the elementary cell $T[R, V_i]$ of the selected register R ,
 20 indicated by the $(i+1)$ -th quartet V_i of the header stored in the buffer memory 15, is read at step 21. If the command field $CT(C)$ of this cell is not 001 (test 22), the contents of this command field are applied to the controller 16 at step 23 so that it will adopt, if
 25 necessary, the status changes needed to proceed with the corresponding actions, after which the module 13 determines how the analysis should continue by identifying the position in the buffer memory 15 of the next quartet to be analysed at step 24 and the next
 30 register of the TRIE memory which should be used to continue the analysis at steps 25 to 28.

At step 24, the code of the next quartet to be analysed is calculated as a function of the code of the current quartet i , the position code represented by the
 35 fields FM and DP of the cell in the TRIE memory and

optionally by the offset parameter OFFSET managed by the controller 16. In accordance with the conventions outlined above, f as a function will be expressed as follows:

```

5      f(i,00,DP,OFFSET) = i + 1;
      f(i,11,DP,OFFSET) = DP;
      f(i,10,DP,OFFSET) = i + DP, where DP is positive or
      negative; and
      f(i,01,DP,OFFSET) = OFFSET + DP, where DP is
10     positive or negative.

```

At step 25, the module 13 examines the field CP of the current cell of the TRIE memory in order to determine whether a count is necessary. If a count is not necessary (CP=00), the register designated by the pointer read from the field Ptr of the cell is selected as being register R for the subsequent iteration at step 26. If CP(C)≠00 at step 25, the address A of the counter to be incremented is obtained at step 27 as being the contents of the field Ptr of the current cell of the TRIE memory. At step 28, the module 13 then issues a command for this counter V(A) to be incremented and retrieves the pointer PT(A) in order to select the register R for the following iteration.

This analysis process continues by linking several logical analyses supplying intermediate statuses. It is terminated when the value 001 appears in the command field CT(C) of the current cell (test 22). At this point, the module 13 feeds to the controller 16 the parameters contained in the fields TT and PA of the cell at step 29, then applies the final status Ptr(C) to the routing module 12 at step 30.

C L A I M S

1. A method of associating routing references to data packets by means of a TRIE memory (14) by analysing in succession, from different carrier registers of the TRIE memory, different portions of a header of each packet containing protocol data, characterised in that as a packet arrives, its header is stored in a buffer memory (15) and a first portion of the stored header is analysed, in that each analysis of a header portion of a packet produces either the forwarding reference associated with the packet or an intermediate reference containing a first code (FM, DP), making it possible to locate at an arbitrary location of the buffer memory a next portion to be analysed, and a second code (Ptr), which makes it possible to locate, at an arbitrary location of the TRIE memory, a carrier register from which said next portion is to be analysed and in that, having analysed the first portion of a stored header, the subsequent portions are analysed in accordance with the first and second codes contained in the intermediate references produced in succession until the forwarding reference is produced.

2. A method as claimed in claim 1, in which the TRIE memory (14) consists of elementary cells, each having a field (CT) indicating whether the cell contains a forwarding reference and in which each elementary cell which does not contain a forwarding reference has at least a first field (FM, DP) to contain a piece of information designating a location of the buffer memory (15) at which the analysis is to be continued and a second field (Ptr) to contain a piece of information designating a register of the TRIE memory with which the analysis will be continued.

3. A method as claimed in claim 2, in which said information designating a location of the buffer memory (15) at which the analysis is to be continued contains a format code (FM) indicating the numerical representation of said location of the buffer memory and an address code (DP) designating said location of the buffer memory in accordance with said representation.

10 4. A method as claimed in claim 3, in which the format code (FM) indicates (i) a sequential representation whereby the data elements of the stored header are analysed in the order in which they appear, (ii) an absolute representation whereby the next data element to
15 be analysed is designated by its absolute position in the stored header, (iii) a differential representation whereby the next data element to be analysed is designated by its position relative to the data element currently being analysed or (iv) a relative
20 representation whereby the next data element to be analysed is designated by its position relative to a given portion of the stored header.

5. A method as claimed in any one of claims 2 to 4, in
25 which each elementary cell which does not contain a forwarding reference has an additional field (CC) which contains a piece of information indicating whether a counting operation is necessary and in which each elementary cell which does not contain a forwarding
30 reference and whose additional field (CC) indicates that a counting operation is necessary has, in its second field (Ptr), a pointer (A) to a memory location (V(A)) containing a counter to be incremented when said operation is performed and associated with another
35 memory location (PT(A)) containing a pointer (R) to the

register of the TRIE memory (14) with which the analysis will be continued.

5 6. A method as claimed in any one of claims 1 to 5, in which each elementary cell of the TRIE memory (14) has a command field (CT) which contains instructions destined for an auxiliary controller (16).

10 7. A method as claimed in claim 6, in which a forwarding reference is issued from the TRIE memory (14) with parameters applied to the controller (16) so as to specify any amendments to be applied on a portion of the header of the packet stored in the buffer memory (15).

15 8. A packet router (10) having a circuit (13) for analysing the header of packets received, using an associative memory of the TRIE type (14) operating in accordance with a method as claimed in any one of the preceding claims.

A METHOD OF ASSOCIATING ROUTING REFERENCES WITH
DATA PACKETS BY MEANS OF A TRIE MEMORY AND
A PACKET ROUTER APPLYING THIS METHOD

Different portions of a header of each packet containing protocol data are analysed in succession from different carrier registers of the TRIE memory. As a packet arrives, its header is stored in a buffer memory (15) and a first portion of the stored header is analysed. Each analysis of a portion of header produces either the routing reference associated with the packet or an intermediate reference containing a first code, making it possible to locate at an arbitrary location of the buffer memory a next portion to be analysed, and a second code, making it possible to locate at an arbitrary location of the TRIE memory a carrier register from which this next portion is to be analysed. Having analysed the first portion of a stored header, the subsequent portions thereof are analysed in accordance with the first and second codes contained in the intermediate references produced in succession until the routing reference is produced.

Figure 3

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.